Real-time multi-stage deep neural network control for SCExAO

B. POU (1,2,3), V. DEO (4), K. AHN (4), S. VIEVARD (4), J. LOZI (4), O. GUYON (4,5,6,7), E. QUINONES (1), M. MARTIN (2), D. GRATADOUR (3)

(1) Barcelona Supercomputing Center (BSC), (2) Universitat Politècnica de Catalunya (UPC), (3) Observatoire de Paris, (4) Subaru Telescope, (5) Steward Observatory, (6) Wyant College of Optical Sciences [University of Arizona], (7) Astrobiology Center of NINS
Introduction

Motivation
- Extreme AO requires exquisite phase reconstruction and prediction in real-time
- Streamline AO pipeline (e.g., no need for gain selection)
- Experiment in a realistic environment

Proposal
- Develop plugin to handle real-time streams with ML and to connect to telescope software
  - TensorRT
  - MILK
- Test new ML methods on the SCExAO bench
Subaru Telescope and SCExAO

**Subaru telescope**
- 8.2 m diameter
- Altitude: 4139 m
- Hawaii, USA

**SCExAO**
- 50x50 Deformable Mirror
- Visual Pyramid Wavefront Sensor
- **Hardware available**
  - AMD EPYC 7763 64-Core Processor (x2). 3.1 GHz, Max 3.5 GHz
  - GPUs: A6000, RTX 3080 Ti (x2), GPU 2080 Ti (x2)

---

*Source: Subaru Telescope webpage*
Methods
Methodology: multi-stage ML

Non-Linear Reconstruction – Supervised Learning
• P-WFS: non-linear relationship between P-WFS image and phase
• Close the loop even under strong non-linear behaviour

Predictive Control – Reinforcement Learning
• Correct for error introduced by the system delay and atmospheric evolution
Non-linear reconstruction: U-Net

Training
- Supervised Learning
- Gather data from the bench pushing modes/actuators randomly -> Generalize to any distribution
- U-Net with 8 layers
- L1 relative loss
Predictive Control: Reinforcement Learning

Correct for **temporal error** with Deep Reinforcement Learning (RL)

- **Trial and error**
- **Policy** ($\pi$): given **state** predict **action** that maximises cumulative **reward**
  - State ($s_t$): reconstruction and history of reconstruction/history of commands
  - Reward ($r_t$): drives the policy to minimise future reconstructions
  - Action ($a_t$): residual command
- **Model-free RL**
- **Online training**
Pipeline

Predictive Control

Postprocessing

Filter

Integration

To DM

Training

RTC4AO - B. POU

8
Simulation results (COMPASS)

- 40x40 DM, 56x56 P-WFS
- Various atmospheric conditions

Differences with real-world

Ensure real time performance
MILK: **Modular Image processing Library toolKit**

Framework for high-performance image processing with shared memories (SHM)

- **Main components:**
  - **Function Parameter Structure (FPS):** Reading/writing parameters of processes
  - **Process information (ProcInfo):** Process management
  - **Streams**
  - **Modularity**

**Diagram:**

- **SHM modifyable by any process**
- **Streams**
- **Real-time Process**
- **SHM linked to process**
- **Process Info**
- **Function Parameter Structure**
Deep neural network plugin for MILK

- Integrate MILK with high performance computing library TensorRT
  - Offline training models
  - Online training models

**Code structure**

![Code structure diagram]

**Offline training models**

![Offline training models diagram]
Deep neural network plugin for MILK II

Online training models

**Process 1**

Thread 1

Input Stream → Preprocessing → Model 1 → Postprocessing → Output Stream

Thread 2

Wake up thread → Copy weights to Model 2 → Swap model 1 and 2 → Sleep thread

**Process 2**

Python

Train for N steps → Copy → Train for N steps → ...

C/C++

(No hard real-time)

(No hard real-time)

From ms to sub-ms!
Results
RL results: bench

Loop parameters:
- 2 Khz
- Num. modes controlled = 500
- PWFS r_{modulation} = 100 mas
- Bright star

Atmos parameters:
- A_{atmos} = 0.2 \mu m
- Wind speed = 20 m/s
- Reduced amplitude on lower order modes (to simulate first stage)

No fitting error
U-Net results: bench (same parameters)

**Example 1**

- Rec. (non-lin): 0.074 um RMS
- Turb.: 0.065 um RMS
- Res. (non-lin): 0.029 um RMS

**Example 2**

- Rec. (non-lin): 0.127 um RMS
- Turb.: 0.167 um RMS
- Res. (non-lin): 0.081 um RMS

Tip tilt removed
Real-time results: RL + U-Net

- Loop Frequency: 1 KHz
- Hardware:
  - Using 1 GPU A6000
  - Using 2 GPU RTX3080 Ti
  - 1 GPU RTX2080 Ti
  - 1 CPU Core per process

Critical Path

- Average over 40 K frames
  - acqWFS: 958 µs
  - U-Net: 731 µs
  - MVM: 212 µs
  - RL pipeline: 227 µs

Could run at around 1 kHz
Real-time: Issues

**Jitter**

**Time per update (RL)**

- Time per update: 0.08 ms
- Total updates to learn from scratch: 20K
- Total time: 26 min
- Still, RL won’t be learning from scratch all the time
Conclusion

SCExAO is a unique environment to test new ideas in a state-of-the-art 8m telescope
1. Demonstrated potential of new ML methods on the SCExAO bench (off-real time)
2. Highlighted the extra difficulties in real-life compared to a simulation
3. Implemented a library for real-time inference of deep neural networks which is integrated into RTC software – to be released soon ...
4. Once the constraints are solved, I expect to try it on-sky with SCExAO

Integration into COSMIC
Questions?

Feel free to email me at bartomeu.poumulet@bsc.es